P7B.12

EXPERIMENTS WITH A SELF-ADAPTIVE MULTIGRID BAROTROPIC TROPICAL CYCLONE TRACK MODEL

Scott R. Fulton*, Nicole M. Burgess, and Brittany L. Mitchell

Clarkson University, Potsdam, New York

1. INTRODUCTION

Accurate prediction of hurricane tracks may require re-solving the flow both within and around the storm. Since the spatial scales in these two regions differ substantially, uniform resolution is inherently inefficient: the grid should be refined only near the storm. This paper describes the performance of a barotropic model with an adaptive multi-grid scheme which automatically refines the mesh around the storm as it moves.

2. MODEL DESCRIPTION

We formulate the model on a section of the sphere using a Mercator projection (true at latitude $\phi = \phi_c$). The model consists of the modified barotropic vorticity equation

$$\frac{\partial \zeta}{\partial t} + m^2 J(\psi, \zeta) + \beta m \frac{\partial \psi}{\partial x} = \nu m^2 \nabla^2 \zeta, \qquad (1)$$

where the relative vorticity ζ and streamfunction ψ are related by

$$L\psi := \left(\nabla^2 - \frac{\gamma^2}{m^2} \right) \psi = \frac{\zeta}{m^2}. \qquad (2)$$

Here $\nabla^2 = \partial^2/\partial x^2 + \partial^2/\partial y^2$, $J(\psi, \zeta)$ is the Jacobian of (ψ, ζ) with respect to (x, y), $\beta = 2\Omega \cos \phi / a$ (with a and Ω the radius and rotation rate of the earth), and $m = \cos \phi_c / \cos \phi$ is the map factor. There are two quasi-physical parameters: the diffusion coefficient ν, and the parameter γ (inverse of the effective Rossby radius) which helps prevent retrogression of ultralong Rossby waves. The model domain is a rectangle in x and y centered at $(x, y) = (0, 0)$, where $(\lambda, \phi) = (\lambda_c, \phi_c)$. At the boundaries we specify the streamfunction ψ (and thus the normal component of the velocity); where there is inflow, we also specify the vorticity ζ.

The space discretization uses second-order finite differences on uniform rectangular grids (as explained below), approximating the advection terms by the Arakawa Jacobian. The time discretization uses the classical fourth-order Runge-Kutta (RK4) scheme; this is highly accurate, allows relatively large time steps for stability, and—since it is a one-step scheme—is easy to implement and has no spurious computational modes. These discretizations are embedded in an adaptive method which includes local refinements in both space and time as detailed in the next section.

---

*Corresponding author address: Scott R. Fulton, Dept. of Mathematics and Computer Science, Clarkson University, Potsdam, NY 13699–5815; email: fulton@clarkson.edu.

3. ADAPTIVE MULTIGRID METHOD

To adapt the resolution near the storm we superimpose nested uniform grids with different mesh sizes. The base grid G_1 covers the the entire computational domain, while successively finer patches G_2, G_3, ... are strictly nested as shown in Fig. 1. The basic time stepping algorithm is similar to that used in most nested models (e.g., Berger and Oliger, 1984). Specifically, with two grids (coarse and fine) one full time step is executed as follows:

1. One step (length Δt) on the coarse grid,
2. Two steps (length $\Delta t/2$) on the fine grid, with boundary values interpolated from the coarse grid in space and time,
3. Transfer the fine-grid solution to the coarse grid where they overlap.

This algorithm generalizes recursively to more than two grids. Each time step consists of four RK4 stages, each of which involves predicting ζ from (1) and diagnosing ψ from (2).

To solve for ψ on any computational grid G_l we use a multigrid method. With local time stepping as above, the next coarser grid G_{l-1} may be at a different time t, so the multigrid method usually requires using an additional set of coarse grids covering the same domain as G_l. If these "local coarse grids" are used at all times, the resulting method includes only one-way interaction between the grids: the fine grid uses boundary values from the coarse grid, but its effect is only felt on the coarse grid through the solution transfer in the overlap region at the end of the time step.
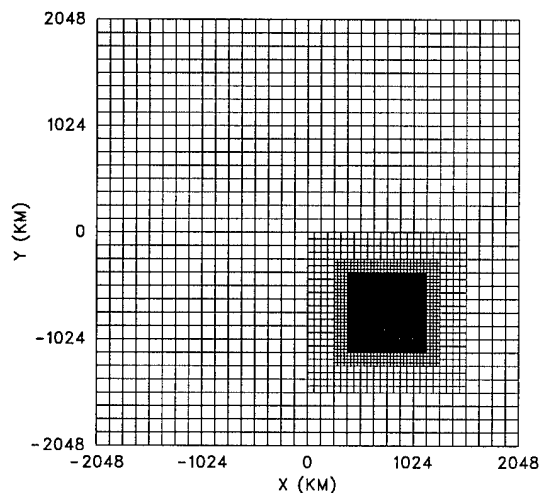

Figure 1. Computational grids ($h = 128$, 64, 32, 16 km).

The local coarse grids must be used in the first time step on the fine grid G_l in the Berger-Oliger algorithm. However, in the final RK4 stage of the second fine-grid time step—when the next coarser computational grid G_{l-1} is at the same time level—the full computational grid G_{l-1} may be used instead of the first local coarse grid in the multigrid cycling. In this way, fine-grid information can affect the entire coarse grid immediately through the relaxation process, so the multigrid algorithm automatically achieves two-way interaction between the computational grids. Note that this approach requires Full Approximation Scheme (FAS) processing, since each grid patch covers a smaller domain than the previous one. Specifically, if we denote the discrete approximation to (2) on the fine grid G_l with mesh size $h = h_l$ as

$$L^h \psi^h = F^h, \tag{3}$$

then the corresponding equation to be solved in the overlap region on the coarse grid G_{l-1} with mesh size $2h = h_{l-1}$ is

$$L^{2h} \hat{\psi}^{2h} = \hat{F}^{2h} := L^{2h}(\hat{I}_h^{2h} \tilde{\psi}^h) + I_h^{2h}(F^h - L^h \tilde{\psi}^h), \tag{4}$$

and the corresponding correction to the fine-grid solution is given by

$$\tilde{\psi}^h \leftarrow \tilde{\psi}^h + I_{2h}^u(\hat{\psi}^{2h} - \hat{I}_h^{2h} \tilde{\psi}^h). \tag{5}$$

Here, $\tilde{\psi}^h$ represents the approximate solution for ψ on the fine grid, the fine-to-coarse transfer operators $\hat{I}_h^{2h}$ and I_h^{2h} represent injection and full weighting, respectively, and the coarse-to-fine transfer operator I_{2h}^h represents bilinear interpolation. In the region of the coarse grid not covered by the fine grid, the FAS equation (4) is replaced by $L^{2h} \psi^{2h} = F^{2h}$ and no correction (5) is needed.

4. SELF-ADAPTIVITY

In previous versions of the model (e.g., Fulton, 1997) the grid sizes were fixed in advance, and simply moved when necessary to keep the location of the vorticity maximum roughly centered on each patch. Figure 2 shows sample results quantifying the gains in efficiency available with the mesh refinement strategy given above. Each point gives the mean track error over 72 h for a single model run; the label gives the finest mesh size (in km) and size of the patches, if any (side length as a fraction of the full domain width, with A=1/2, B=3/8, C=1/4, D=3/16, and E=1/8). For example, the grids in Fig. 1 are for the run 128BCD. The gain in efficiency due to mesh refinement varies significantly, with the best combinations of patches giving speedups in the range 10–20. Since the optimal combination of patch sizes will depend on both the vortex and the surrounding flow and may vary as the solution evolves, a fully self-adaptive procedure would be advantageous. Here we introduce such a method, using truncation error estimates to decide automatically where to refine or coarsen the mesh.

The necessary truncation error may be estimated essentially for free using FAS processing as described above. Specifically, the relative truncation error

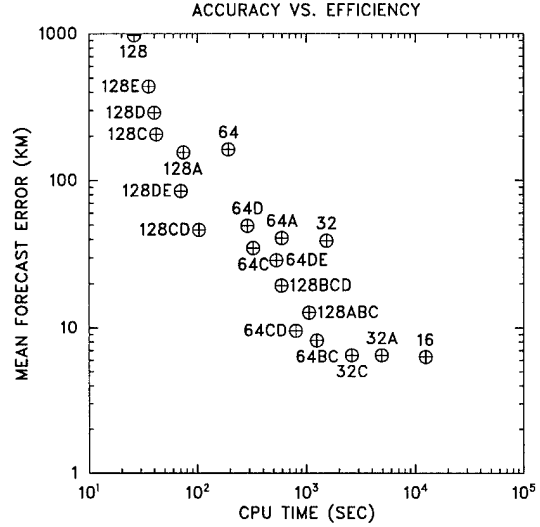$$\tau_h^{2h} = \hat{F}^{2h} - \hat{I}_h^{2h} F^h \tag{6}$$

Figure 2. Accuracy vs. efficiency (from Fulton, 1997)

provides a fourth-order approximation to the difference in truncation error between grids h and $2h$. Thus, the truncation error on grid h can be approximated with accuracy $O(h^4)$ as $\tau^h \approx \frac{1}{3} \tau_h^{2h}$. This estimate is computed at each point on grid h at the end of the time step. Where τ^h is small, the current fine grid h is no longer needed, so its extent may be reduced. Conversely, where τ^h is large, further refinement may be beneficial, so these points are flagged and a new finer grid $h/2$ enclosing them is introduced. The determination of what is "small" or "large" follows the algorithm outlined in Brandt (1977, sec. 8), which uses a parameter λ to control the trade-off between increased accuracy and increased work. Note that since F^h and $\hat{F}^{2h}$ are already computed in the solution process, computing the truncation error estimate requires very little additional work. Also, the estimated truncation error can be used in an extrapolation method to obtain fourth-order accuracy from the second-order discretization. We will present results at the conference quantifying the performance of the self-adaptive method outlined above.

REFERENCES

Berger, M., and J. Oliger, 1984: Adaptive mesh refinement for hyperbolic partial differential equations. *J. Comp. Phys.*, **53**, 484–512.

Brandt, A., 1977: Multi-level adaptive solutions to boundary-value problems. *Math. Comput.*, **31**, 333–390.

Fulton, S. R., 1997: A comparison of multilevel adaptive methods for hurricane track prediction. *Elec. Trans. Num. Anal.*, **6**, 120-132.